



Juego de Adivinación

Transcripción

[00:00] Continuando con nuestro curso de lógica de programación, lo que vamos a hacer en esta oportunidad es un nuevo programa que es un juego de adivinación. Vamos a reaprovechar nuestro código del IMC, IMC mejorado que lo hicimos en la anterior aula. Para ello, vamos a venir aquí, save as, guardar como, y lo vamos a nombrar Juego de Adivinación. Perfecto.

[00:37] Vamos a eliminar todo lo que no necesitamos, hasta aquí. ¿Qué es lo que nuestro juego va a hacer? Bueno, le vamos a pasar un mensaje, un mensaje al usuario que ingrese un número entre 0 y 10. Vamos a configurar, fijar un número aquí en sistema, y él, en el primer intento le vamos a decir al usuario si acertó o si erró en su intento de acertar el número que estaba programado en el sistema.

[01:15] Entonces, para ello, vamos a crear una variable que la vamos a llamar número pensado o número del sistema, que vamos a ponerle 5, que ven ahí en la mitad, y var, otra variable que sea número lanzado. Entonces, ese va a ser el número que le vamos a pedir a nuestro usuario. Ya sabemos que es un número y sabemos que la función prompt de Java nos devuelve un texto.

[01:54] Entonces, ya lo vamos a convertir a int entero, a número, y le vamos a decir a nuestro usuario: "Ingrese un número entre 0 y 10". Okay. Tenemos los dos números. Ahora, ¿qué hacemos con estos dos números? Los tenemos que comparar y tenemos que decirle al usuario: "Usted acertó" o "usted erró". ¿Cómo? Aplicando las condicionales que ya hemos aprendido.

[02:26] Entonces: si, una condición, va a ser verdadero. Y otro si, la segunda condición, caso el tipo no acierte, caso nuestro usuario no acierte. Entonces, número pensado es igual a número lanzado, si el número pensado es igual al número lanzado, el número que está configurado en la máquina es igual al número que el usuario va a ingresar, ¿qué es lo que hacemos? Lanzamos un mensaje.

[03:03] Ya tenemos nuestra función imprimir que "usted acertó". Perfecto. Si el usuario no acertó, vamos a decirle imprimir "usted erró" y le vamos a inclusive dar una ayuda: "El número pensado era" de lo concatenemos, número pensado. Aquí vamos a reaprovechar esta parte de nuestro código. Solo que aquí no es igual, es diferente que así se define la comparación de diferente en JavaScript.

[04:00] Hay algunos programas de programación que usan este parámetro, esta forma de decir que son diferentes, que estamos diciendo: "esto es diferente de esto". En JavaScript es el signo de admiración para abajo e igual. Vamos a guardarlo. Vamos a irnos de aquí. Bueno, este es el IMC, lo vamos a cerrar, ya no lo vamos a usar.

[04:27] "Ctrl + O" Juego de Adivinación. Abrimos nuestro programa y aquí nos dice: "Ingrese un número entre 0 y 10". 3. Usted erró, el número pensado era 5. El número configurado era 5. "Ingrese un número". 9. 5. Usted acertó. Okay, nuestro programa funciona, sin embargo, es un programa poco inteligente. En el primer intento, ya el usuario sabe que el número era 5.

[05:02] Y una vez acertó, siempre va a acertar. El juego de adivinación acaba cuando la persona acierta la primera vez. Imagínense si este juego estuviera en una reunión de amigos, en una confraternización de amigos y quieran jugar esto, el juego va a acabar cuando descubran que el número que está configurado es un 5.

[05:26] ¿Cómo podemos hacer para que este número sea variable, para que sea un número aleatorio que sea dado por la máquina, que no dependa de un humano para configurar este número? Existe la función `random`, de `Math` en JavaScript. Para mostrarles y demostrar esta función, vamos a ir a nuestro ambiente desarrollador, "Ctrl + Shift + I" o venir aquí, Más Herramientas y Herramientas de Desarrollador.

[05:58] Aquí lo tengo en portugués, por eso dice `ferramentas`. Bueno, tenemos como les decía, dentro del módulo de `Math`, de matemáticas, tenemos una función que se llama `random`. Esa función me devuelve un número entre 0 y 1, un número aleatorio que el sistema calcula por sí solo entre 0 y 1.

[06:27] Para repetir la última función que yo coloqué, pueden apretar flecha para arriba, y fíjense que cada vez es un número diferente, nunca va a repetir ese número o va a ser muy difícil que repita, en realidad. Voy a volver aquí. Entonces, `Math.random`. Enter. Fíjense que me da un número entre 0 y 1: 0.58. Ahora, ¿cómo hago para que este número sea un número entre 0 y 10? Porque me devuelve un número entre 0 y 1.

[07:03] Muy fácil, lo único que tengo que hacer es multiplicarlo por 10. Y ahora, lo que me devuelve es un número aleatorio entre 0 y 10 con una gran cantidad de números decimales. Pero yo quiero solo un número entre 0 y 10, sin las casas decimales. ¿Cómo puedo hacer eso?

[07:30] Uso otra función que está dentro del módulo de `Math`, que es `round`. Abro y cierro paréntesis y dentro de aquí coloco mi función `random`. Función `random`, abro y cierro paréntesis, porque esto es una función, recordemos que las funciones tienen que abrir y cerrar paréntesis, y aquí lo multiplico por 10. Lo voy a multiplicar y lo voy a redondear sin posiciones decimales que por defecto este `round` me devuelve.

[08:05] ¿Puedo definir las casas decimales? Sí. Pero para este ejercicio lo quiero el número entero. Entonces, 3, 2, 6, y está funcionando. Entonces, lo que voy a

hacer es llevar esta función a nuestro programa y voy a sustituir este 5 por eso.

[08:31] Entonces, lo que estoy haciendo es: "Calculame un número randómico", le estoy diciendo a JavaScript: "Calculame un número randómico, multiplícalo por 10 para que esté entre 0 y 10 y redondealo para que me devuelva un número entero entre 0 y 10". Luego la lógica es la misma: voy a compararlo con el número que va a ingresar el usuario y le voy a decir si él acertó o erró.

[08:56] Ahora, el programa está mucho más divertido. A ver, vamos a probarlo, voy a cerrar esto, voy a actualizar aquí. "Ingrese un número del 0 al 10". 8. "Usted erró, el número pensado era 5." De nuevo, vamos a ver si efectivamente está funcionando o es un 5 clavado. No, está funcionando, y dos veces que ya fue 5. Vamos a seguir intentando de nuevo. 7. "El número pensado era 5."

[09:32] 4. Vamos a ver por qué. Bueno, aquí tenemos, primeramente, no es un error, pero nos olvidamos cambiar el título, entonces vamos a poner programa juego de adivinación, ¿correcto? Aquí lo estamos calculando, número pensado. Ahora, ¿por qué este número pensado? Vamos a guardarlo y aquí vamos a actualizar 4. Ahora sí. Programa juego de adivinación, el número pensado era 3.

[10:29] De nuevo. 9. Era 7. F5, 7. Era 5. 5, era 2. Vamos a ver si estamos con suerte. Estamos 0 suerte, no acertamos ni una. Usted acertó, ahí acertamos con el 6. Perfecto, entonces, aquí acabamos con los intentos. Entonces, nuestro programa está funcionando, ya es un programa divertido, yo puedo jugar con esto, faltan todavía bastantes mejoras que podemos hacerle.

[11:19] Por ejemplo, limitar el número de intentos puede ser, y limitar también, condicionar el ingreso del usuario, porque hoy en día el usuario puede ingresar el número 500 y puede compararlo con el resultado de la máquina y el número 500 nunca va a aparecer porque lo estamos limitando aquí, ¿okay?

[11:38] Sin embargo, lo que podemos mejorar ya ahora en nuestro código es, que en estos casos, que tenemos dos condiciones, tenemos dos caminos en

este árbol de decisión. Si el número pensado es igual al número lanzado, significa que acerté. Caso contrario, si no, si yo no acerté, significa que erré. Java me permite usar un comando que es else, que significa si no, si pasa esto, haga esto.

[12:10] Si no, si no entró aquí, ya directamente que haya la siguiente instrucción, ya no necesito validar la condición. Lo único que necesito es decirle que me ejecute una instrucción caso no entre aquí. O sea, if, número pensado es igual a número lanzado, que imprima. Caso contrario, que imprima que erró, porque él o acierta o erra, no hay una tercera opción.

[12:44] Recordemos un poco del ejercicio del IMC, donde calculábamos el índice de masa corporal, ahí teníamos cuatro condiciones. Se puede aplicar el else, pero en esa opción sería un poco más complicado. ¿Concuerdan conmigo? ¿Están de acuerdo? Porque al tener cuatro condiciones sería más difícil colocar el else. Aquí tenemos solo dos, no necesito colocarle si acerté, si erré.

[13:13] Directamente, si acertó, haga esto. Caso contrario, haga la siguiente instrucción, que es lo que estamos haciendo aquí. Las características del if y del else son exactamente las mismas. Todas las instrucciones que estén dentro de las llaves van a suceder, lo mismo con el else.

[13:32] Vamos a guardarlo, vamos a ir a nuestro código para probarlo, actualizamos. Ingrese un número: 7. Era 1. F5, 8, era 7. 9. Era 4. Vamos ahí, acertamos. Entonces ya creamos un juego.

[0:14:00] Este juego vamos a continuar personalizándolo y mejorándolo, haciéndolo más divertido todavía, pero lo importante es que hemos aprendido a calcular números randómicos, que se usan bastante en programación, y continuamos mejorando y explorando el tema de condiciones. Nos vemos en el próximo video. Gracias.

